MSDOS 2.0

by Paul G. Allen

Future plans for MSDOS

or

The Bridge to XENIX

The previous speakers have already described the current state of MSDOS. I am going to talk about MSDOS 2.0, the next major release of MSDOS, as well as areas that we are researching for future releases. We are quite excited about the enhancements we are making and we feel that 2.0 has a number of features that represent the state of the art in operating systems.

Over the last seven years, Microsoft has ported its software products to over fifty different operating system environments. This, combined with our experience with XENIX, has given us a thorough education in what OEMS, end users and programmers want and don't want in an operating system.

I hope I've whetted your appetites a little. So, what is 2.0 MSDOS, when will it be available, and what are the added features?

**** (slide #1) ****

First of all, it is important to realize that MSDOS is part of a family of operating systems.

MSDOS has already been described as Microsoft's single user, single tasking operating system. It is written in 8086 assembler and is very compact.

MSDOS takes about 7K bytes for the 1.1 version kernal as compared with about 50K for a minimal XENIX kernal. XENIX has over 5 megabytes of utilities (compilers, assemblers, text processors, etc.) and really should be used with a hard disk. MSDOS, on the other hand, fits comfortably with all its utilities on two floppy disks. Providing the user with a family of operating system capabilities means a clear migration path from MSDOS to XENIX. This means compatibility for both the terminal end user and the systems programmer.

MSDOS 2.0 will be available to current MSDOS OEMS at no charge in the third quarter of this year.

**** (slide #2) ****

Here is a list of the enhancements to be added to MSDOS for the 2.0 release. The enhancements emphasize greater user friendliness, standardization, XENIX compatibility features, networking, improvements to the standard utilities as well as the addition of some common XENIX "filters", and improved disk performance.

I will go through each of these areas in detail as we proceed.

**** (slide #3) ****

The Visual Shell

The end user interface or "shell" is the first thing that the user sees when he boots MSDOS. The shell interprets all commands the user types to the operating system. MSDOS 2.0 replaces the traditional command line oriented shell with a visual shell that shows the user a menu of the most commonly executed applications and utilities.

The screen is divided into different "categories" that are in separate windows on the screen. The user may then use cursor movement keys to step between categories and also select a particular program in a category. If the user types the "?" key, then "help" information will be displayed for the category or the program. The help information is context sensitive. For instance, if the cursor was positioned at the "copy" command, information on copying files would be shown. Also displayed are the current date and the time of day. At the bottom portion of the screen there is a window which contains part or all of the directory for the currently selected disk drive. Cursor positioning may be used to select a file in a similar fashion as a program was selected. This saves the user from having to type the file name, as well as reducing the possibility of selecting the wrong file.

One very important feature is that the user may customize the shell to his own needs. He may create his own categories, programs, and help files. This could be used to tailor MSDOS for a particular applications environment, or for use in a foreign country.

Foreign language support also extends to the operating system error messages. For example, an OEM could change the messages to French or Kangi for the Japanese market.

Menus, context sensitive help files, and the ability to customize menus are also standard features of the Microsoft Multi-Tool family. This use of a consistent end user interface is very important.

XENIX and the Microsoft SoftCard will also support visual shells that are identical in nearly every respect.

**** (slide #5) ****

Program and Driver Interface Standards

Program transportability is of key importance for an operating system like MSDOS. Programs written on one machine should be transportable to another similar, although not identical machine. In particular, many of the latest applications programs like Multiplan, word processing packages, or the visual shell, need to access the terminal of the system in a standard way both for screen output and input.

MSDOS provides two kinds of standard interfaces. The first is a standard interface for applications programs that want to talk to the keyboard and screen in a standard way so they do not have to be customized for the different environments that MSDOS will provide. This is done by using ANSI standard escape sequences in the terminal character stream for input of cursor movement keys and for writing text and to the screen.

The operating system intercepts the ANSI sequences and converts between the terminal's actual cursor movement character codes and the ANSI ones. Such an approach does not in any way restrict the kind of terminals that can be used. MSDOS 2.0 will also support AT&T's presentation level text and graphics protocol. The presentation level protocol is a resolution independent method of describing in a serial byte stream, characters, and graphics objects. This means that a graphics image may be stored on disk just as if it had been written to the screen. The standard also is device independent with respect to device resolution and other parameters. A program that drew a chart on the screen could just as easily send the same image to a printer without changing the program, just the device name used.

Of course, another benefit of using the presentation level protocol is that an MSDOS system could interface directly to a cable network or database system that used the protocol with the addition of a small program to receive the incoming character stream. For both the ANSI sequences and the presentation level protocol, Microsoft will supply an example driver that can be customized for different terminals.

**** (slide #6) ****

The burden of supporting the standard protocols falls on the device drivers, where it belongs. This means that the code required to perform graphics functions and other special I/O need be written only once, not rewritten for each application program.

MSDOS 2.0 can also load device drivers dynamically. The drivers are loaded when the system is booted and a new driver is incorporated by running a simple utility and then re-booting. Manufacturers of peripheral equipment can supply the installable drivers with their hardware, so the user will have true configurability of his system.

Each device has an associated logical name such as "lpt1". The user could replace the standard printer driver with a new one, say for a Centronics instead of an Epson, and then re-boot the system. Or, he could have two printer drivers, "lpt1" and "lpt2". Space for the drivers is allocated dynamically. As Bob O'Rear said earlier, being able to access a device as a file makes it easy to take advantage of a new driver without having to add code to the application.

**** (slide #7) ****

Besides the already mentioned compatibility of the visual shell and some of the new utilities I will discuss later, there are more places that XENIX compatibility has been built into MSDOS 2.0. New system calls are provided to perform I/O functions in the same fashion as XENIX. Besides the standard system calls that already exist for support of stream I/O, there will be system calls that expand or contract the amount of memory allocated to a program, and parse a command line.

A standard library for XENIX-86 C will allow compilation of a program on a XENIX system and then execution on MSDOS. This will allow MSDOS to tap the already existing library of programs written in C, as well as the generation of new utilities which can run under either XENIX or MSDOS.

Standard input, output and error files are passed to applications programs in the conventional XENIX form, and they may be re-directed to files, or used to "pipe" the output of one utility to another. Piping means sending the output of one program to the input of another program. In the case of MSDOS, the actual piping is done with a temporary disk file. One of the strengths of XENIX is that it allows modular system utilities or "filters" as they are sometimes known. These filters can be piped together to perform such functions as sorting, searching, modifying or generating files. Then, instead of building a specific program to perform a new function, the already existing filters and utilities may be tied together to perform a new function, instead of building the functionality into a new program. A trivial example might be the output of the directory command to a sort utility.

The command shell may be left resident, and re-invoked from inside a program. This means, for instance, that a directory could be listed by invoking the shell inside an application program instead of having to build a directory listing capability into every application.

A "XENDOS" utility will be provided with XENIX-86 to allow execution of MSDOS programs under a XENIX environment. This will enable applications written in 8086 assembler for MSDOS to be immediately transported to XENIX-86 if desired. As Chris Larson described earlier, most applications can be converted to XENIX merely by recompiling them in the high level language they are written in.

**** (Slide #8) ****

Networking

Networking is a key to the success of operating systems like MSDOS and XENIX in the office automation market.

An enhancement package to MSDOS will provide local network capability. Microsoft's networking software will encompass both XENIX and MSDOS. An advanced mail system, file transfer program and other utilities will sit on top of the basic network services provided by the respective operating systems.

XENIX systems will be able to function as network file servers, and MSDOS systems as application servers for individual users. Using a XENIX system as a file server will mean that the XENIX node can handle the simultaneous I/O requests of many MSDOS workstations, with the potential to support use as an applications node itself.

Microsoft plans to support a standard network protocol, and we are currently examining both the DoD IP/TCP standard as well as the XEROX NS protocols used with Ethernet. It is our view that the higher level protocol chosen for a network system is much more important than the medium used. In other words, whether a network system uses twisted pair, an Ethernet

cable, a broadband network or a fiber optic cable does not matter as much as the kinds of functions that the protocol can provide.

Independance of messages from particular hardware and support of network booting and broadcasting are just a few of these functions.

We plan to prototype our network on IBM Pcs to iron out the bugs and demonstrate the capabilities of our software.

Microsoft has a complete in-house mail system right now that is used by all our executives and managers. This system runs under XENIX, and has been in use for some months already. We are also networking three internal XENIX systems together, a VAX and two 11/70's, using Ethernet hardware and software from 3COM.

**** (slide #9) ****

Improved Utilities

There will be a number of improved utilities available with 2.0 that were not included in previous releases of MSDOS. The printer spooler will contain a buffer of data to be printed and will use keyboard wait time to interrupts to send characters to the printer. The spooler will have a queue of files to be printed that may be changed by a utility while the printer spooler is running.

An enhanced DEBUG program will allow typein of assembler mnemonics.

Do *** While and If *** Then *** Else features in Batch to test the completion codes from the compilers.

A number of XENIX compatible utilities (grep, sort, etc.) These are the filters I mentioned earlier.

Improved line editor (EDLIN) with features such as block copy, better defaulting of parameters, and merge.

**** (slide #10) ****

Better Performance

Disk performance will be enhanced through the use of multiple disk buffers. XENIX already has this feature. Effectively, it treats a portion of memory as a "cache" for disk requests. A counter is kept for each buffer which indicates how recently it was accessed. When a new sector must be read, a check is made to see which buffer is the "oldest", in other words has been used least recently. This buffer is then written out to disk if it has been written into. Then the new sector replaces the slot previously occupied by the old one.

When MSDOS 2.0 boots, it will determine how many buffers to allocate from a parameter stored in a system call on disk. The buffers will be forced out on a least-recently-used basis. Thus the user can change the cell to suit his applications demands for speed of disk access versus main memory utilization.

We think that with the price of memory as low as it is, most users will allocate a reasonably large number of buffers, in most cases perhaps memory equivalent to two tracks.

Ramdrive

Finally, virtual disks implemented in RAM may become an integral part of many users systems. Actually this is really just an example of another kind of device driver, with RAM mimicing the functions of a floppy or hard disk. Microsoft will shortly release a product for MSDOS that implements such a pseudo disk in a painless way. The resulting ten-fold speed increase can dramatically reduce the time to compile and link programs. Of course the speed of a RAM drive may be used in many other ways.

**** (slide #11) ****

What does the future hold for MSDOS after 2.0? Here are a few of the more important features to follow.

We plan to re-code MSDOS in C to make in portable, like XENIX, so that it may be moved to new microprocessors quickly. Of course, once written in C, this version will also include a C compiler generating native code for the micro that it runs on.

Two features of networking that we plan to exploit are very important. It is very desirable to make files appear as if they are independant of their location. In other words, it shouldn't matter if a file is local or remote. This is crucial for real distributed data base systems.

Another important aspect of network systems of the near future will be the appearance of "diskless" workstations. The cost of a network interface will be less than the cost of providing a local hard disk. This means that if the proper software is there, it will be more economical to boot the operating system and run with a remote file-server acting as local storage than to put a hard disk in each machine. The high performance of networks like Ethernet and Datapoint's Arcnet and Corvus' OMNINET makes this entirely practicable and this has already been done.

Microsoft will continue to provide many enhancements to the basic BASIC compiler, PASCAL, FORTRAN and COBOL compilers which already run under MSDOS. These changes will go hand in hand with the improvements to MSDOS itself. As you can see, the next year will be one of rapid evolution for MSDOS. We think that with the changes and upgrades we have planned, MSDOS will become the premier single user operating system. Thank you.

SLIDE 1

The Operating System

In the beginning, the technological world of man was null, void and without form, and man said, "Let there be Computers !" ... and lo there appeared hardware.
And man said, "Let there be life in these computers !" ... and alas there was none.
Man then said, "Let there be software!" and lo and behold there suddenly appeared life ...

SLIDE 2

The Role of The Operating System

- To provide an interface between the application program and the hardware system and its peripherals.
- Alternatively to provide the application with a set of primitives for communications with the hardware.
- To assure standards for applications programs communications with the hardware.
- Provide mechanisms for error recovery.
- To afford simple interfaces for application programs authors.

SLIDE 3

The Importance of Choosing the Right Operating System

- Assures transportability of software between machines of the same generation and from generation to generation.
- Provides the widest selection of applications packages
- Affords maximum explotation of hardware capability
- Allows more sophisticated applications to be supported
- Simplifies application development
- Reduces applications development costs
- Provides an environment for high level languages
- Operating system maintained by those who recognize this as a serious responsibility and one that is not to be exploited.

SLIDE 4

CP/M-80 the 8-Bit Standard ???

- CP/M-80 became a standard not because of its features or sophistication but rather because of its availability on a wide range of hardware.
- In the applications environment the operating system is not typically a highly visible part of the system.
- De facto standards are not always true standards
- There are currently approximately a dozen CP/M-80 compatible operating systems (several with enhanced features).

Important Operating System Features

- Provide a good set of primatives for authors
- User friendly
- Good error recovery procedures
- Capable of handling large files and disks
- Easy to implement on a wide variety of systems
- Internal blocking/deblocking
- Device independent I/O

Operating System Standards

SLIDE 6

- Standards are an important part of the microcomputer future.
- Applications programs have a significant role to play in the development and maintenance of such standards.
- Authors will actively support such standards